(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 May 2016 (26.05.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/081880 A1

(51) International Patent Classification:
G06F 7/06 (2006.01)

(21) International Application Number:
PCT/US2015/061949

(22) International Filing Date:
20 November 2015 (20.11.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/082,918 21 November 2014 (21.11.2014) US
62/109,203 29 January 2015 (29.01.2015) US

(71) Applicant: **TRUSTEES OF BOSTON UNIVERSITY** [US/US]; One Silber Way, Boston, MA 02215 (US).

(72) Inventors: **CASTANON, Gregory, D.**; 65 Child Street, Jamaica Plain, MA 02130 (US). **SALIGRAMA, Venkatesh, D.**; 48 Summer Street, Newton, MA 02459 (US). **ZHANG, Ziming**; 1 Emerson Place, Apt. 6L, Boston, MA 02114 (US). **CHEN, Yuting**; 438 Old Connecticut Path, Framingham, MA 01701 (US).

(74) Agents: **SUNSTEIN, Bruce, D.** et al.; Sunstein Kann Murphy & Timbers LLP, 125 Summer Street, Boston, MA 02110 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: LARGE SCALE VIDEO SEARCH USING QUERIES THAT DEFINE RELATIONSHIPS BETWEEN OBJECTS

(57) **Abstract**: One embodiment of the present solution is a relationship-based video server system. The server system is programmed to provide a platform for large scale video search based on relationships between objects in video segments, and the platform executes search based on probabilistically discriminative objects and relationships.

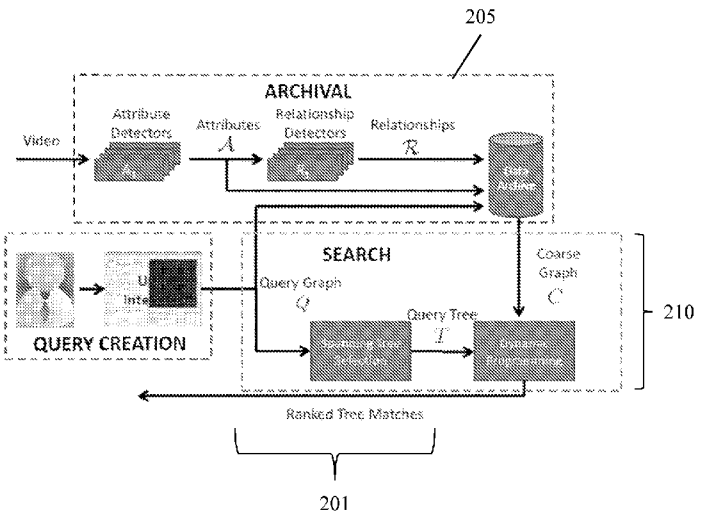


FIG. 2

WO 2016/081880 A1

LARGE SCALE VIDEO SEARCH USING QUERIES THAT DEFINE RELATIONSHIPS BETWEEN OBJECTS

Related Art

[0001] The present application claims the benefit of U.S. Application No. 62/082,918, entitled "Large Scale Video Search Using Queries that Define Relationships Between Objects" and filed on November 21, 2014, and U.S. Application No. 62/109,203, entitled "Large Scale Video Search Using Queries that Define Relationships Between Objects" and filed on January 29, 2015, which have been hereby incorporated by reference in their entireties.

Federally Sponsored Research

[0002] This invention was made with Government Support under Contract No. N00014-13-C-0288 awarded by the Office of Naval Research. The Government has certain rights in the invention.

Technical Field

[0003] The present disclosure relates to searching large volumes of videos, and more particularly, to finding videos whose content includes objects and relationships between the objects that most closely match the objects and relationships between objects in a user's graphical video search query.

Background Art

[0004] It is known in the prior art to tag training data with learning classifiers and to train video search systems based on the tagged data. However, these video search systems require significant amounts of training data to function properly, and tagging training data can be time and process-intensive endeavor.

Summary of the Embodiments

[0005] One embodiment of the present solution is a relationship-based video server system. The server system is programmed to provide a platform for large scale video search based on relationships between objects in video segments, and the platform executes search based on probabilistically discriminative objects and relationships. The server system is configured to run processes that, for each video segment received for storage in a video database, analyze image data in the video segment to recognize objects that are represented in the video segment and identify relationships between the objects. Additional processes determine, for each video segment, at least one of the most discriminative object or the most discriminative relationship, based on probabilities of occurrence for the recognized objects and identified relationships. Additional processes determine, for each video segment, a data structure with the recognized objects and identified relationships of the video segment and that is also based on at least one of the most discriminative object or the most discriminative relationship. Additional processes store the data structure for each video segment in the video database.

[0006] Additional processes receive a video search query defining relationships between a plurality of objects. Additional processes determine, based on probabilities of

occurrence for the objects and relationships of the video search query, at least one of the most discriminative object or the most discriminative relationship of the video search query. Additional processes identify, based on at least one of the most discriminative object or the most discriminative relationship of the video search query, a subset of the data structures to search for video segments. Additional processes determine, for each video segment represented in the subset of the data structures, a score based on similarity between the objects and the relationships in the video search query and the objects and the relationships associated with the video segment. Additional processes display representations of video segments with the highest scores.

[0007] Additional processes may, for each video segment received for storage, analyze the image data in the video segment to identify at least one attribute of at least one recognized object. Additional processes may, for each video segment received for storage, determine a set of discriminative objects and relationships based on probabilities of occurrence for the recognized objects and identified relationships. Additional processes may, for each video segment received for storage, configure the data structure for the video segment by hashing the recognized objects and identified relationships of the video segment. Additional processes may, for each video segment received for storage, configure the data structure for the video segment by hashing the objects and the relationships in the set of discriminative objects and relationships. Additional processes may, for each video segment received for storage, stored the data structure for the video segment in a fuzzy hash table.

[0008] Additional processes may receive a video search query in a graphical representation, wherein each object is represented by a unique node and each relationship is represented by a unique edge. Additional processes may determine a set of discriminative

objects and relationships for the video search query. Additional processes may determine a maximally discriminative spanning tree based on the set of discriminative objects and relationships for the video search query.

[0009] Additional processes may identify the subset of the data structures to search by downsampling the stored video segments based on at least one of the most discriminative object or the most discriminative relationship of the video search query. Additional processes may identify the subset of the data structures to search by downsampling the stored video segments based on the set of discriminative objects and relationships for the video search query. Additional processes may identify the subset of the data structures to search by creating a graph based on the maximally discriminative spanning tree of the video search query and the stored video segments. The graph may be a tree structure. In some embodiments, at least one relationship in the video search query is a spatial relationship.

[0010] Additional processes may determine, for each video segment represented in the subset of the data structures, the score based on similarity by computing subscores between the objects in the video search query and the objects in the video segment, computing subscores between the relationships in the video search query and the relationships in the video segment, and summing the subscores. Additional processes may determine, for each video segment represented in the subset of the data structures, the score based on similarity that accounts for tolerance levels of a user.

[0011] Additional processes may predict a user intent of the video search query, determine at least one potential alteration to the video search query based on the user intent, and display an altered video search query as a proposed search query for the user.

[0012] In some embodiments, each video segment has a duration of a pre-determined interval, and the duration may be 15 minutes.

[0013] One embodiment of the present solution is a computer program product having a non-transitory computer-readable medium with computer code thereon for searching large scale video data. The product includes program code for receiving a video search query defining relationships between a plurality of objects. The product includes program code for determining, based on probabilities of occurrence for the objects and relationships of the video search query, at least one of the most discriminative object or the most discriminative relationship of the video search query. The product includes program code for accessing a video database that stores video segments. The product includes program code for identifying, based on at least one of the most discriminative object or the most discriminative relationship of the video search query, a subset of the video segments stored in a video database to be subject to search. The product includes program code for determining, for each video segment in the subset subject to search, a score based on similarity between the objects and the relationships in the video search query and the objects and the relationships associated with the video segment. The product includes program code for displaying representations of video segments with the highest scores.

[0014] In some embodiments, the product includes program code for determining a set of discriminative objects and relationships for the video search query.

[0015] Further embodiments of the present solution include an apparatus for searching large scale video data, and a computer program product having computer code thereon configured to search large scale video data, when executed. In some embodiments, the product includes program code for receiving the video search query in a graphical

representation, wherein each object is represented by a unique node and each relationship is represented by a unique edge. In some embodiments, the product includes program code for downsampling the stored video segments based on at least one of the most discriminative object or the most discriminative relationship of the video search query. The apparatus and computer program product both encompass the various features described herein.

Brief Description of the Drawings

[0016] The foregoing features of embodiments will be more readily understood by reference to the following detailed description, taken with reference to the accompanying drawings, in which:

[0017] FIG. 1 is an exemplary flow diagram for searching large volumes of video data using queries that define relationships between objects;

[0018] FIG. 2 is an exemplary flow diagram for creating a representation of a video search query based on its objects and relationships, creating representations of stored videos based on the objects and relationships in their contents, and finding videos whose objects and relationships most closely match the objects and relationships of the video search query;

[0019] FIG. 3 is an exemplary flow diagram for creating a tree structure corresponding to the objects and relationships from a graphical video search query;

[0020] FIG. 4 is an exemplary graphical video search query input by a user, which depicts objects and defines relationships between the objects;

[0021] FIGS. 5 and 6 are exemplary tree structures for the video search query of FIG. 4, wherein the tree structure in FIG. 5 is more discriminative than the tree structure in FIG. 6;

[0022] FIG. 7 is an exemplary flow diagram for representing videos in a manner that facilitates efficient search and retrieval;

[0023] FIG. 8 is an exemplary flow diagram for searching for videos that most closely match the user's video search query;

[0024] FIG. 9 is an exemplary process for creating a representation of video content that may be used for efficient searching;

[0025] FIG. 10 is an exemplary process for searching for videos whose content best matches the video search query, and

[0026] FIG. 11 is an exemplary computing device that may be used to implement the invention.

Detailed Description of Specific Embodiments

[0027] Modern video surveillance systems amass large volumes of video content. For example, an airport may include numerous cameras positioned throughout its facilities, each one recording content on an on-going basis. Likewise, companies may position cameras in the parking lots of retail stores, and government agencies may set up cameras to monitor vehicular activity on highways. While these surveillance systems provide an enormous wealth of data, simply by the sheer volume of video content collected, searching the video content remains problematic.

[0028] For example, when a user searches videos for a particular activity, that activity may be expressed in a variety of ways. Suppose a user wants to search for an accident involving a vehicle. Although the collision may involve two or more vehicles, the collision may occur between one vehicle and a pedestrian or a bicyclist. Further, the vehicles

may be cars, buses, trucks, or motorcycles. In this manner, the activity "vehicular collision" may be manifested in numerous forms. However, when users search videos, a keyword-based system may eliminate many potential matches from being found. In another example, some surveillance systems may extract objects (e.g., people, cars) from videos, and store the same. However, such systems analyze videos based on the temporal ordering of the objects contained therein, and this approach's ability to accurately describe activities is limited.

[0029] Various embodiments enable users to formulate video search queries that account for spatial relationships between objects, which permits videos to be searched for particular activities with greater accuracy. A user may specify objects that the user wishes to see in the videos and may describe the relationships between the objects. In this manner, users may create video search queries in a more intuitive manner, and using the information contained therein, the video search system may retrieve higher quality matches among the stored videos. Furthermore, the user may specify the extents to which the search results may deviate from the particular video search query. As a result, the video search system can search for videos in a more accurate manner while retaining potential search results that deviate from the user's exact query.

[0030] In general overview, FIG. 1 depicts an exemplary flow diagram for searching large volumes of video data using queries that define relationships between objects. First, a video search system receives a video search query that includes objects and relationships between the objects (step 1001). For example, the video search system may provide a graphical user interface that enables a user to construct a graphical video search query. From this interface, the user may specify the objects that the user wants in the video search results. In this manner, the user may require a search result to include a "person", a "car", and a

"package", by way of example. In another example, the user may require a search result to include a "person", a "dog", and a "car". Moreover, the user may specify characteristics of any given object. For example, the person may be a particular person (e.g., John Smith), the car may be red, and the package may be large.

[0031] In some embodiments, the user may select an object from a menu, such as a drop down menu, and position the object on the interface. Each object that the user chooses becomes a "node" (also referred to herein as "vertex") in the video search query.

[0032] From the graphical user interface, the user may also specify the relationships between the objects, and each relationship becomes an "edge" in the video search query. For example, a user may draw a line between two objects to indicate that the objects are related to one another and then specify the relationship. In various embodiments, the user may select a type of relationship from a menu and associate this relationship with the line connecting two objects. For example, a user may draw a line between "person" and "dog", select "near" from a menu of relationship types, and associate "near" with the line. In another example, a user may draw a line between "person" and "car", select "riding in" from a menu of relationship types, and associate "riding in" with the line.

[0033] In some embodiments, the video search system provides a user interface that receives video search queries as text queries. The user interface may include a single box that receives text. The user may input the objects required to be in the video, such as "person", "car", and "package", and define the relationships between the objects.

[0034] In some embodiments, the user may describe the requirements of the video search query, and the video search system may process the query to extract the objects and their relationships. The video search system may use natural language processing, or any

other methods, for parsing the text input. For example, the user may input "A person is driving a car at about 35 mph, and a dog is sitting in the passenger seat", and the video search system may determine that the query includes a person, a car, and a dog. The system also determines that person and the dog are both positioned within the car, and that the car, person, and dog are all moving at about 35 mph. In some embodiments, the video search system may reformulate the user's input into a search query that the system can use for further processing, according to the processes described herein.

[0035] Further, the user may input the video search query as a string that uses Boolean or Boolean-like terms. In some embodiments, the user interface displays a list of relationships, such as "near", "moving", "still", "riding in", "disappears", that may be used for video search queries. Thus, the user may input a query like (car MOVING "35 mph") and (person RIDING IN car) and (dog RIDING IN car). In another example, a user might input a search query for a person carrying an object to a car and putting it in the car trunk, as (person NEAR package) and (person NEAR car) and (package DISAPPEARS). As before, the video search system may reformulate the user's input into a search query that the system can use for further processing, according to the processes described herein.

[0036] After the video search system receives the video search query, the video search system generates a representation of the video search query (step 1003). Thus, the video search system may transform the video search query into a structure conducive for searching database(s) of videos.

[0037] In one example, the video search system may convert the video search query into a maximally discriminative query. When the user inputs a text string, the system may extract the objects and their relationships and then use statistics based on individual objects,

relationships between particular objects, or both, to reformulate the query. Using the statistics, the system may determine which objects or relationships occur most infrequently. Then, the system may create a query that highlights these objects or relationships. In this manner, the system would create a search query that minimizes the number of videos needed for further consideration to determine if the videos match the search query's criteria.

[0038] In another example, the video search system may convert the video search query into a tree structure. Because the same graphical video search query may be represented by different tree structures, the video search system determines which tree structure highlights the most distinguishing characteristics of the search query (also referred to herein as the "maximally discriminative spanning tree"), to reduce the number of possible videos for initial and/or further consideration.

[0039] Then, the video search system determines, for each video in a plurality of videos, a score based on similarity between the objects and relationships of the video search query and objects and relationships of content in the video (step 1005). When the videos were first received for storage in the database(s), the video search system extracted objects, information about the objects themselves, and the relationships between the objects from the video content. This information was further processed and stored in a manner conducive to searching, which will be described in more detail below.

[0040] The video search system compares the objects and relationships from the user's video search query with the objects and relationships of video content, and assigns scores to videos based on the level of similarity, as will be described in more detail below. Thus, the video search system does not search for exact matches between the user input and the stored videos. Instead, the video search system still considers videos that have objects or

relationships that are similar, rather than exactly the same, compared to those in the user query. However, the video search system may assign lower component scores for objects or relationships that are merely similarly. In this manner, the more a video's content deviates from the search query, the lower its score, but the video search system may still consider it.

[0041] Finally, the video search system selects the videos with the highest scores (step 1007). In various embodiments, the video search system may select ten, or any other number, or videos with the highest scores, and display thumbnails for the videos to the user. Alternatively, the video may automatically begin playing the video with the highest score, and arrange the remaining videos as a prospective playlist.

[0042] FIG. 2 depicts an implementation of the video search system described in FIG. 1. In particular, FIG. 2 depicts an exemplary flow diagram for receiving a video search query and creating a corresponding representation 201, creating representations of stored videos based on the objects and relationships in their contents 205, and finding videos whose corresponding objects and relationships most closely match the objects and relationships of the video search query 205. Creating a representation of a video search query 201 will be described in more detail in reference to FIGS. 3-6, creating representations of the objects and relationships of the contents of stored videos 205 will be described in more detail in reference to FIG. 7, and finding videos whose corresponding objects and relationships most closely match the objects and relationships of the video search query 205 will be described in more detail in reference to FIG. 8.

[0043] In some embodiments, to create a representation of a video search query, as described in FIG. 3, the video search system first receives a graphical video search query from a user (step 3001). As previously discussed, a user may create a query by positioning

nodes, corresponding to objects, on the interface and connecting the nodes with edges, each edge defining the relationship between the respective objects.

[0044] In some embodiments, the user may also specify characteristics of the objects (also referred to herein as "attributes"). The user may select an attribute from a menu on the graphical user interface and associate the attribute with an object. For example, the user may indicate the color or size. The user may specify a general direction of motion of an object, such as "moving from left to right", or the approximate speed of motion, such as "moving between 45-60 mph". Alternatively, the user may require the object to be at rest, e.g., "no motion". In another example, the user may indicate that one object has a constant presence in the video, or that the object disappears from view. Furthermore, the user may specify a particular identity for an object, such as requiring a person to be "John Smith" or a car to be a "sedan".

[0045] The user or the video search system may identify images, video, or other multi-media that depict exemplary relationships between objects. For example, the user may upload images or videos that provide examples of what particular relationships, such as "near", connote to the user. In another example, the user may upload a video that demonstrates the speed at which a user wants an object in target video to move. The video search system may analyze the relationship in the provided media and use this data as the reference point when determining whether videos in the database(s) include objects that are have the relationships required by the user's query. In some embodiments, the video search system may access existing databases of images or videos that depict the relationships, and extract information about the relationships to use as reference points. Moreover, the video

search system may search databases over computer networks, such as the Internet, to find multi-media that include such relationships.

[0046] In various embodiments, the user may specify tolerance levels for deviations from the attributes. For example, the user may manually input a tolerance level for an attribute. After indicating that an object should be three (3) feet wide, the user may also indicate that objects between 2.5-3.5 feet wide may also be considered. Likewise, the user may wish to search for a red object, but because colors may vary depending on the amount of light in a scene, pink objects may be acceptable candidates for a match. Although the user may wish to look primarily for dogs, the user may also permit search results that include cats instead. In another example, after requiring an object to be still, the user may nevertheless allow the object to move between 1-5 mph, in case an external object causes the object of interest to move by accident. In some embodiments, the user may specify tolerance levels, and in alternate embodiments, the video search system may store predefined tolerance levels for any attribute displayed in the graphical user interface. Moreover, the user may configure the video search system to default to predefined tolerance levels, or to enable users to input tolerance levels that override the predefined ones.

[0047] The user may also specify relationships between the objects. The user may select a relationship from a menu on the graphical user interface and associate the relationship with two objects. For example, the user may indicate that two objects are near one another, two objects move in tandem with one another, or that one object is being carried by another. In other examples, the user may indicate that one object is less than five (5) feet away from a second object, or that two objects are moving towards one another.

[0048] The user or the video search system may identify images that depict the objects of interest. For example, the user may upload images of a person, such as "John Smith". The user may upload images of the particular car of interest in his or her search query. In some embodiments, the user may upload video or other multi-media content that depicts the objects that are referenced in the search query. The video search system may extract the objects and their attributes from the provided data, and use this data as the reference point when determining whether videos in the database(s) include objects that are similar to those in the user's query.

[0049] Similar to attributes, the user may specify tolerance levels for deviations from the selected relationships. For example, after indicating that two objects should be less than five (5) feet from one another, the user may also indicate that objects separated by 5-7 feet may also be considered. For a search query in which two objects are moving towards one another, the user may also indicate that the objects may stop moving at approximately the same time (e.g., thereby, indicating a collision).

[0050] An exemplary graphical video search query which depicts objects and defines relationships between the objects is shown in FIG. 4. In this example, the user is creating a query for a person loading a package into a car. Since the query requires three objects - the person, the car, and the package - a user will insert three nodes, each corresponding to one of the objects, into the search query. Since all three objects must be near one another for the person to load the package, the user may draw edges between the objects to define the "near" relationship between all of them (e.g., the triangle in the right part of FIG. 3). Also, the package disappears from view after the person loads the package into the car, and this characteristic may be incorporated by assigning the "end" attribute to the package node (e.g.,

"Object end"). Additionally, to make the search query more specific, the user may require that the person be proximate to the package before it disappears. Thus, the user may add additional nodes for the person and package, require the person and package to be proximate to one another, and require this continued proximity when the package disappears (e.g., the "person" and "object" nodes on the left part of FIG. 4).

[0051] Additionally, the video search system may analyze the inputted search query to predict the user's intent and make suggestions for the search query. In some embodiments, the system stores commonly occurring objects and relationships. As the user inputs a search query, the system may recognize that the user is creating a particular relationship between objects, and display a recommended search query.

[0052] After the user finishes constructing a graphical video search query, the video search system may store information from the query. In some embodiments, the video search query may be represented as:

[0053] $VSQ = Q(V^q, E^q, A^q, R^q, L^q)$

[0054] Through this disclosure, the superscript "q" is used to indicate that the variables pertain to the user's video search query.

[0055] In this representation, V^q represents the objects in the user input, with each object corresponding to a node in the query. In the notation, the representation of a node may also be referred to as a "vertex". The *i*-th object would be represented by V^q_i.

[0056] A^q represents the characteristics for each of the nodes. Thus, the characteristics for the *i*-th node would be represented by A^q_i.

[0057] L^q represents the tolerance levels for the characteristics of the nodes. Thus, the tolerance levels for the characteristics for the *i*-th node would be represented by L^q_i.

[0058] E^q represents the edges indicating that vertices are related. Thus, for all pairs of related nodes v_1 and v_2, $(v_1, v_2) \in E^q$

[0059] R^q represents the particular relationships between nodes. Thus, for each pair of related nodes v_1 and v_2, $R^q_{(v1, v2)}$ represents the set of relationships between the two nodes (v_1, v_2).

[0060] L^q_v represents the tolerance levels for the relationships. Thus, for each pair of related nodes v_1 and v_2, $L^q_{(v1, v2)}$ represents the tolerance levels for the relationships between the two nodes (v_1, v_2).

[0061] After the user inputs the video search query, the video search system creates at least one tree structure corresponding to the video search query (step 3003). In various embodiments, the video search system selects one of the nodes to serve as a tree's root, and creates the remainder of the tree based on the nodes and relationships of the video search query relative to the root node. By way of example, FIGS. 5 and 6 depict two tree structures that both correspond to the video search query of FIG. 4. In FIG. 5, the tree structure 500 is rooted at "Object end", whereas in FIG. 6, the tree structure 600 is rooted at "Person".

[0062] Furthermore, the video search system determines the maximally discriminative spanning tree ("MDST") (step 3005). The maximally discriminative spanning tree highlights the most distinguishing (e.g., the least common) characteristics of the search query. By way of example, the tree structure 500, rooted at "Object end" which is connects to the "car" node via a "near" relationship, is more discriminative than the tree structure 600, which is rooted at "Person". Because videos with objects that disappear near cars are far less common than videos that have a person, a tree structure rooted with the less common node with its associated attributes will be more discriminative than tree structures rooted at more

commonly occurring nodes. Thus, when the tree structure 500 is used as a basis for search, compared to the tree structure 600, non-matching videos will be eliminated from consideration more quickly and the overall search will require less processing time. Thus, selecting a highly discriminative tree structure advantageously minimizes the processing time of the search.

[0063] To evaluate the discriminativeness of tree structures, the video search system may use the probabilities of occurrence for the nodes and relationships. In some embodiments, the video search system stores probabilities regarding attributes for nodes and relationships between nodes, which may be denoted as $P_m(a)$ and $P_m(r)$, respectively. These probabilities may correspond to the percentage of videos within the database(s) that include the corresponding attributes for the node or relationship between nodes.

[0064] Any given tree may be represented by:

[0065] $T = T(V^t, E^t, A^t, R^t)$, with a root at T_o

[0066] Through this disclosure, the superscript "t" is used to indicate that the variables pertain to a tree that corresponds to the user's video search query.

[0067] The probability that any given node in the tree will appear in videos in the database is estimated by the product of the probabilities of the attributes for the node:

[0068] $$P_m(v) = \prod_{a \in A_v^t} P_m(a)$$

[0069] Similarly, the probability that any given set of relationships between two nodes in the tree will appear in the videos is estimated by the product of the probabilities of each relationship in the set:

$$P_e(v_1, v_2) = \prod_{r \in R_{(v_1, v_2)}} P_m(r)$$

[0070]

[0071] Suppose P_i represents the path from the root node T_o to the *i*-th node in the tree, T_i. Then, the probability that a video would include the nodes from T_o to T_i may be represented by:

$$\mathcal{P}_m(v_i) = \prod_{v \in \mathcal{P}_i} P_m(v) \prod_{(v_1, v_2) \in \mathcal{P}_i} P_m(v_1, v_2)$$

[0072]

[0073] As a result, the probability that a video would include all of the nodes in a tree T would be:

$$S(T) = \sum_{\mathcal{P}_i \in \mathcal{P}} \prod_{v \in \mathcal{P}_i} P_m(v) \prod_{(v_1, v_2) \in \mathcal{P}_i} P_m(v_1, v_2)$$

[0074]

[0075] The video search system thus seeks a tree structure that minimizes the value of S(T). The probability S(T) is dominated by the root node and the nodes that are nearest to it. As such, the total score of each tree depends disproportionately on the discriminativeness of the nodes and relationships near the root, as their contribution to the value of S(T) is far greater than that of lower nodes.

[0076] In some embodiments, the video search system randomly generates tree structures from the video search query and samples from the created structures. For example, the video search system may use Broder's method to sample from the set of trees. The system may compute the discriminativeness of the sampled trees, and select the tree with minimal S(T) and maximum discriminativeness as the MDST.

[0077] In various embodiments, the video search system determines a maximally determinative query from a user's text input. As described above, the user may input objects

for the video and relationships between the objects, such as (car MOVING "35 mph") and (person RIDING IN car). The video search system processes the string to extract objects and their relationships. Using statistics, such as the stored probabilities regarding attributes for objects and relationships between objects (e.g., the probabilities denoted as $P_m(a)$ and $P_m(r)$), the video search system may determine which objects or relationships occur most infrequently. Then, the video search system may create a search query based on the most infrequent objects and relationships extracted from the user input. Since the search query is based on infrequent occurring objects and relationships, use of the particular search query minimizes the number of videos that the video search system would need to consider to determine if their contents match that of the user's query. In this manner, creating a maximally discriminative search query would reduce the search space within the video database(s).

[0078] Turning now to creating representations of the objects and relationships of the contents of stored videos 205, FIG. 7 depicts an exemplary flow diagram for representing videos in a manner that facilitates efficient search and retrieval. Although the description herein refers to a single video, one of ordinary skill would appreciate that all videos received and stored in the database(s) are subject to the same processing. In general overview, the video search system receives a video (step 7001). Then, the video search system extracts objects and their characteristics from the video (step 7005), and subsequently extracts the relationships between the objects (step 7010). The video search system hashes the objects and the relationships (step 7015) and stores the hashed values in the database (step 7020).

[0079] In some embodiments, the video search system may receive the videos (step 7001) from various surveillance sites. By way of example, the system may obtain videos

from a number of cameras positioned at different locations along a highway, different locations within a parking lot, or different locations within an airport. Each camera may record and store video at its respective location, and transmit the video to the search system. In some embodiments, a camera may transfer video footage in real time. In other embodiments, a camera may store video footage recorded over a predetermined amount of time, and then transfer the video to the search system at predetermined intervals. For example, a camera may transfer 15 minutes worth of recorded video every 15 minutes, and thereafter, begin over-recording its memory with new video content.

[0080] When the video search system receives a video, the system extracts its objects and their characteristics (step 7005). The system may use various object detectors, such as the Viola-Jones detector, to extract items, people, vehicles, and other discrete entities. The video search system may also extract the color and size of objects. Moreover, the video search system may compare pixels between adjacent video frames to determine the direction and/or magnitude of motion of an object, and such comparison may also identify when the object appears or disappears.

[0081] Based on the objects, the video search system extracts the relationships between the objects (step 7010). The system computes the relationships between pairs of objects, and identifies particular relationships to be stored in the database(s). In some embodiments, the video search system only stores discriminative relationships, which are relatively rare compared to other types of relationships. For example, fewer objects will be "near" one another, or be the "same as before", than objects that are "far from" one another. Consequently, relationships such as "near" and "same as before" better distinguish one video

from another than "far from". Thus, when only discriminative relationships are stored, the video search system can search the database(s) for matching videos more efficiently.

[0082] The video search system hashes the objects, their individual characteristics, and the relationships (step 7015) and stores the hashed values in the database (step 7020). The video search system may hash the location of the video with a particular object, the value of the object, and the value of the object's characteristics, and store the values in an inverted index. The system may use standard hashing for the objects themselves. In some embodiments, the system may use locality sensitive hashing for characteristics that may be subject to some variation, such as size and color. Likewise, the video search system may hash the values of the relationships and store them in the index. Moreover, any of the values may be hashed using fuzzy hashing, which enables objects and relationships to be retrieved within tolerance levels of a video search query.

[0083] The video search system uses the maximally discriminative spanning tree (MDST) or the maximally discriminative query to search the database of hashed values to find videos whose content most closely matches the user's video search query. Referring now to FIG. 8, the video search system may first downsample the stored videos based on the user search query (step 8001). Downsampling reduces the number of videos for full comparison against the video search query by eliminating from further consideration the videos with negligible chance of matching the query. In some embodiments, the video search system builds a graph with videos from the database(s) that include at least one object or one relationship from the search query, within the tolerance levels set by the user. Because this resulting graph (also referred to herein as a "coarse graph") is based on the requirements of the video search query, the coarse graph will include all potential video

matches in the database(s), as well as false positives. Nevertheless, the coarse graph greatly reduces the potential search space for the video search system.

[0084] The coarse graph may be represented by:

[0085] $C = C(V^c, E^c, A^c, R^c)$

[0086] Through this disclosure, the superscript "c" is used to indicate that the variables pertain to the coarse graph created from the stored videos and the user's video search query.

[0087] To create the coarse graph, the video search system begins with the objects. The system retrieves the locations of videos that have the objects V^q and their respective characteristics A^q_v. For example, the system may compute a hash code for the object, hash codes for each characteristic of the object, and take the intersection of the bins corresponding to the codes. The system adds a node v_i to V^c for each of these locations and stores the characteristics in A^c_i.

[0088] Additionally, for every relationship between nodes, $(v_1, v_2) \in E^q$, the video search system retrieves the locations of videos that satisfy every relationship in $R^q_{(v1, v2)}$, whose nodes are also in V^c. For each pair of nodes, the system adds an edge to E^c and stores the set of relationships for the edge in R^c. In this manner, the video search system ensures that all videos from the database that have the individual nodes, their characteristics, and their relationships are incorporated into the coarse graph C.

[0089] In some embodiments, the video search system creates a matching graph H from the coarse graph C, which will be searched to find videos. The matching graph H may be represented by

[0090] $H = H(V^h, E^h)$

[0091] In this matching graph H, each node v ∈ T ⊗ C is a tuple denoting a proposed assignment between a node in the maximally discriminative spanning tree T and a node in the coarse graph H. First, the video search system adds matching nodes from C for the root node in T, and then adds matching successive nodes that satisfy both node and edge relationships in the MDST. FIG. 9 illustrates exemplary processes for creating the matching graph, in which τ_v may be a threshold score for a node to be added to the matching graph H and τ_e may be a threshold score for an edge to be added.

[0092] Then, the video search system determines a score, for each video of the downsampled videos, using the maximally discriminative search tree corresponding to the video search query (step 8005). For a prospective video match, the video search system may compute a subscore for a node in the video search query which indicates the extent to which the node, or a similar node, appears in the video. Likewise, the video search system may compute a subscore for a relationship in the video search query which indicates the extent to which the relationship, or a similar relationship, appears in the video. The system may add all of the subscores to calculate an overall similarity score for the video.

[0093] Scoring may be based on kernel functions, which have some tolerance for error. For individual nodes, the similarity score may be the sum of similarities for each attribute belonging to that node. Thus, the score may be represented by:

$$K_v(v_1, v_2) = \sum_{j=1}^{|A|} K_{a(j),\varepsilon(j)}(a_v(j))$$

[0094]

[0095] *a(j)* corresponds to the characteristics of a node

[0096] *l(j)* corresponds to the tolerance level with respect to a characteristic of a node

[0097] For individual relationships, the similarity score may be the sum of the similarities associated with that edge. Thus, the score may be represented by

[0098]
$$K_e(e_1, e_2) = \sum_{j=1}^{|R|} K_{r(j),l(j)}(r_{e_2}(j))$$

[0099] *r(j)* corresponds to the relationships between two nodes

[00100] *l(j)* corresponds to the tolerance level with respect to the relationships between the nodes

[00101] To find prospective matches to the video search query, the video search system may traverse the maximally discriminative spanning tree from leaves to root to determine the optimal solution for each root match. For each leaf node in the MDST, the video search system computes a similarity score. Then, the video search system may merge nodes in the matching graph H into their parent node, keeping the node that has the best score. The system may repeat this process until only the matches in the matching graph H to the root node(s) of the MDST remain.

[00102] For each of these potential root node matches, the video search system computes a similarity score for their associated trees, and in turn, the extent to which the content of the associated video matches the video search query. In some embodiments, the score may be determined by:

[00103]
$$S(Q, C, M) = \sum_{v_q \in V_Q} K_v(v_q, M(v_q)) + \lambda \sum_{(v_{q_1}, v_{q_2}) \in E_Q} K_e(v_{q_1}, v_{q_2}, M(v_{q_1}), M(v_{q_2})) \quad (4)$$

[00104] Then, ranking the scores of the videos, the video search system may retrieve the videos with the highest scores (step 8010).

[00105] Fig. 11 schematically shows a logical view of an exemplary computing system 1100 that may be used with illustrative embodiments of the present invention. A computing system 1100 may have various hardware allocated to it by a system operator, including one or more processors 1194 (e.g., logical CPUs), volatile memory 1195, non-volatile storage 1196, and input and output ("I/O") devices 1197 (e.g., network ports, video display devices, keyboards, and the like). The computing system 1100 may store program code that, when executed by the one or more processors 1194, performs any of the steps described herein. In some embodiments, the server system described herein may be implemented on one computing system 1100, and in the embodiments, the system may be implemented on a plurality of computing systems 1100 that operate in conjunction with one another. The computing system 1100 may accessible using a data network, which may be inclusive of any data network known in the art, such as a local area network ("LAN"), a virtual private network ("VPN"), the Internet, or a combination of these networks, or the like.

[00106] The disclosed apparatus and methods may be implemented as a computer program product for use with a computer system. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable medium (*e.g.*, a diskette, CD-ROM, ROM, or fixed disk). The series of computer instructions can embody all or part of the functionality previously described herein with respect to the system.

[00107] Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices,

and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies.

[00108] Among other ways, such a computer program product may be distributed as a tangible removable medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the network (*e.g.*, the Internet or World Wide Web).

[00109] Of course, some embodiments of the invention may be implemented as a combination of both software (*e.g.*, a computer program product) and hardware. Still other embodiments of the invention are implemented as entirely hardware, or entirely software. The embodiments of the invention described above are intended to be merely exemplary; numerous variations and modifications will be apparent to those skilled in the art. All such variations and modifications are intended to be within the scope of the present invention as defined in any appended claims.

[00110] Although the above discussion discloses various exemplary embodiments of the invention, it should be apparent that those skilled in the art can make various modifications that will achieve some of the advantages of the invention without departing from the true scope of the invention.

What is claimed is:

1. A relationship-based video server system, programmed to provide a platform for large scale video search based on relationships between objects in video segments, the platform executing search based on probabilistically discriminative objects and relationships, wherein the server system is configured to run processes that:

for each video segment received for storage in a video database,

analyze image data in the video segment to (1) recognize objects that are represented in the video segment and (2) identify relationships between the objects,

determine, based on probabilities of occurrence for the recognized objects and identified relationships, at least one of (1) the most discriminative object or (2) the most discriminative relationship in the video segment,

configure, based on at least one of the most discriminative object or the most discriminative relationship, a data structure that includes the recognized objects and identified relationships of the video segment;

store the data structure in the video database;

receive a video search query defining relationships between a plurality of objects;

determine, based on probabilities of occurrence for the objects and relationships of the video search query, at least one of the most discriminative object or the most discriminative relationship of the video search query;

identify, based on at least one of the most discriminative object or the most discriminative relationship of the video search query, a subset of the data structures to search for video segments;

determine, for each video segment represented in the subset of the data structures, a score based on similarity between the objects and the relationships in the video search query and the objects and the relationships associated with the video segment; and

display representations of video segments with the highest scores.

2. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that, for each video segment received for storage, analyze the image data in the video segment to identify at least one attribute of at least one recognized object.

3. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that, for each video segment received for storage, determine a set of discriminative objects and relationships based on probabilities of occurrence for the recognized objects and identified relationships.

4. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that, for each video segment received for storage, configure the data structure for the video segment by hashing the recognized objects and identified relationships of the video segment.

5. The relationship-based video server system of claim 3, wherein the server system is configured to run processes that, for each video segment received for storage, configure the data structure for the video segment by hashing the objects and the relationships in the set of discriminative objects and relationships.

6. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that, for each video segment received for storage, stored the data structure for the video segment in a fuzzy hash table.

7. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that receive a video search query in a graphical representation, wherein each object is represented by a unique node and each relationship is represented by a unique edge.

8. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that determine a set of discriminative objects and relationships for the video search query.

9. The relationship-based video server system of claim 8, wherein the server system is configured to run processes that determine a maximally discriminative spanning tree based on the set of discriminative objects and relationships for the video search query.

10. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that identify the subset of the data structures to search by downsampling the stored video segments based on at least one of the most discriminative object or the most discriminative relationship of the video search query.

11. The relationship-based video server system of claim 8, wherein the server system is configured to run processes that identify the subset of the data structures to search by downsampling the stored video segments based on the set of discriminative objects and relationships for the video search query.

12. The relationship-based video server system of claim 9, wherein the server system is configured to run processes that identify the subset of the data structures to search by creating a graph based on the maximally discriminative spanning tree of the video search query and the stored video segments.

13. The relationship-based video server system of claim 12, wherein the graph is a tree structure.

14. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that determine, for each video segment represented in the subset of the data structures, the score based on similarity by (1) computing subscores between the objects in the video search query and the objects in the video segment, (2) computing

subscores between the relationships in the video search query and the relationships in the video segment, and (3) summing the subscores.

15. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that determine, for each video segment represented in the subset of the data structures, the score based on similarity that accounts for tolerance levels of a user.

16. The relationship-based video server system of claim 1, wherein the server system is configured to run processes that (1) predict a user intent of the video search query, (2) determine at least one potential alteration to the video search query based on the user intent, and (3) display an altered video search query as a proposed search query for the user.

17. The relationship-based video server system of claim 1, wherein each video segment has a duration of a pre-determined interval.

18. The relationship-based video server system of claim 17, wherein the pre-determined interval is 15 minutes.

19. The relationship-based video server system of claim 1, wherein at least one relationship in the video search query is a spatial relationship.

20. A computer program product having a non-transitory computer-readable medium with computer code thereon for searching large scale video data, the computer program product comprising:

program code for receiving a video search query defining relationships between a plurality of objects;

program code for determining, based on probabilities of occurrence for the objects and relationships of the video search query, at least one of the most discriminative object or the most discriminative relationship of the video search query;

program code for accessing a video database that stores video segments;

program code for identifying, based on at least one of the most discriminative object or the most discriminative relationship of the video search query, a subset of the video segments stored in a video database to be subject to search;

program code for determining, for each video segment in the subset subject to search, a score based on similarity between the objects and the relationships in the video search query and the objects and the relationships associated with the video segment; and

program code for displaying representations of video segments with the highest scores.

21. The computer program product of claim 20, further comprising:

program code for receiving the video search query in a graphical representation, wherein each object is represented by a unique node and each relationship is represented by a unique edge.

22. The computer program product of claim 20, further comprising:

program code for downsampling the stored video segments based on at least one of the most discriminative object or the most discriminative relationship of the video search query.

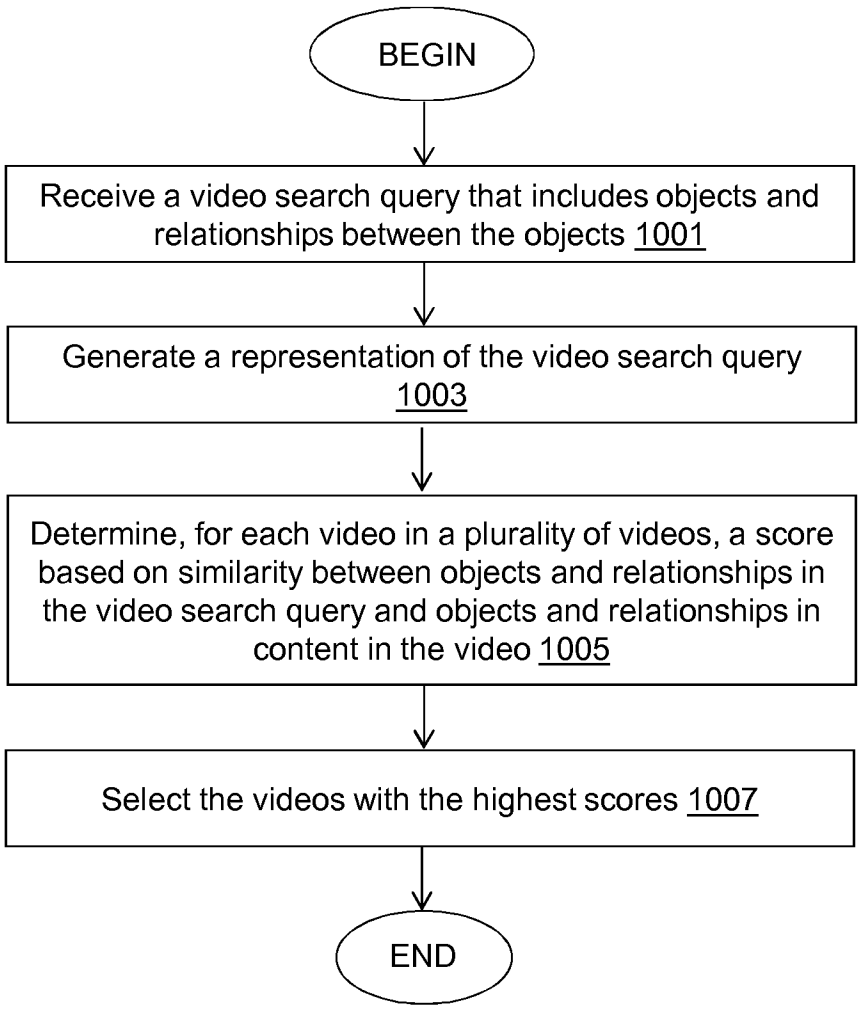

BEGIN
Receive a video search query that includes objects and relationships between the objects 1001
Generate a representation of the video search query 1003
Determine, for each video in a plurality of videos, a score based on similarity between objects and relationships in the video search query and objects and relationships in content in the video 1005
Select the videos with the highest scores 1007
END

FIG. 1

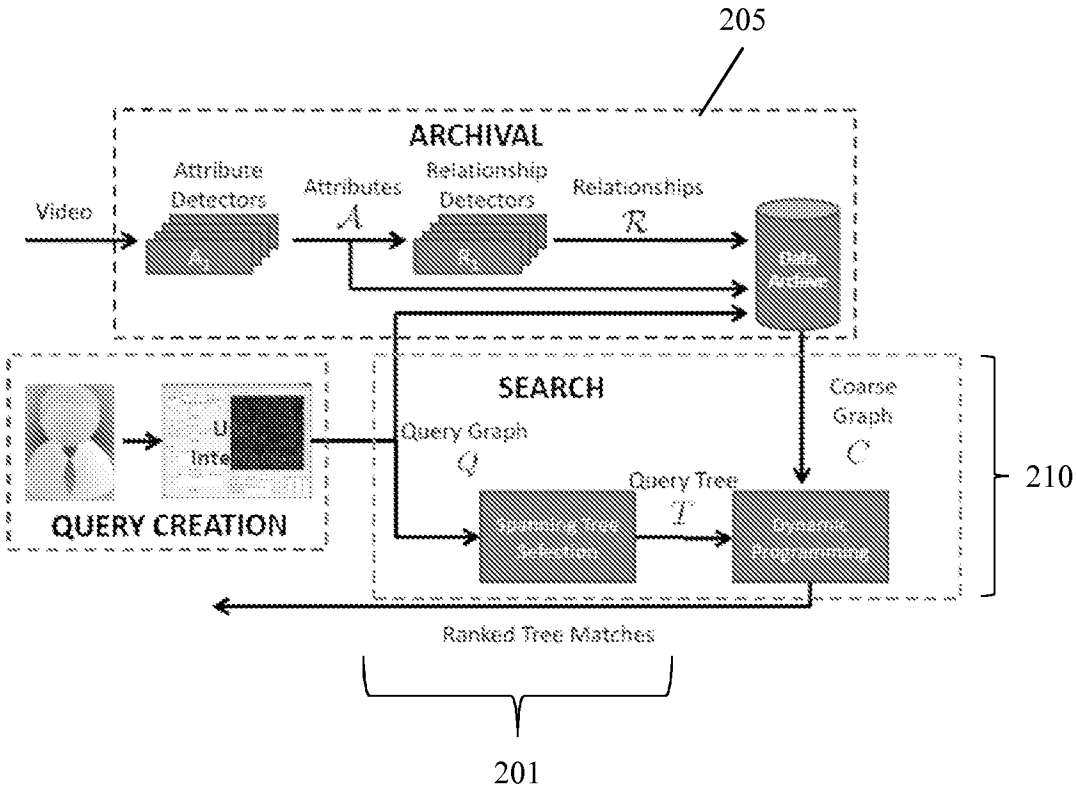

205
ARCHIVAL
Video
Attribute Detectors
Attributes
$\mathcal{A}$
Relationship Detectors
Relationships
$\mathcal{R}$
SEARCH
Coarse Graph
$\mathcal{C}$
Query Graph
$\mathcal{Q}$
Query Tree
$\mathcal{T}$
QUERY CREATION
210
Ranked Tree Matches
201

FIG. 2

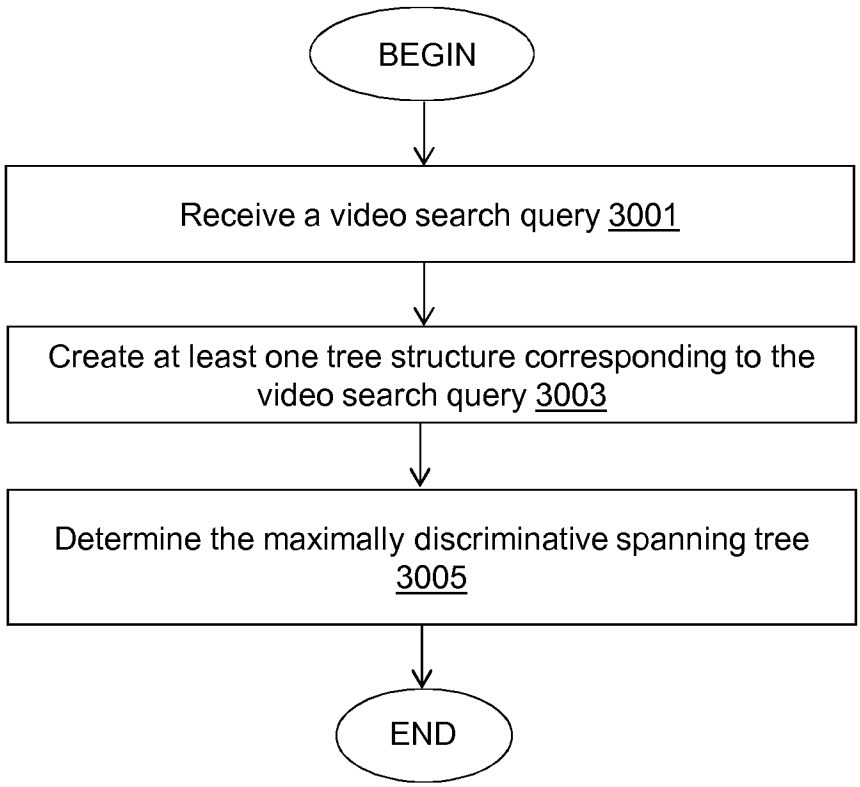

BEGIN
Receive a video search query 3001
Create at least one tree structure corresponding to the video search query 3003
Determine the maximally discriminative spanning tree 3005
END

FIG. 3

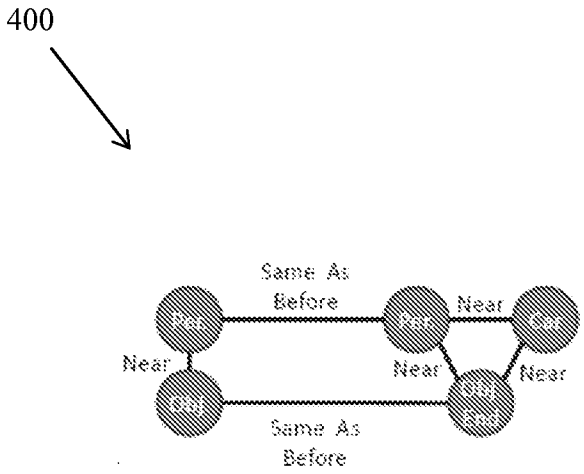


FIG. 4

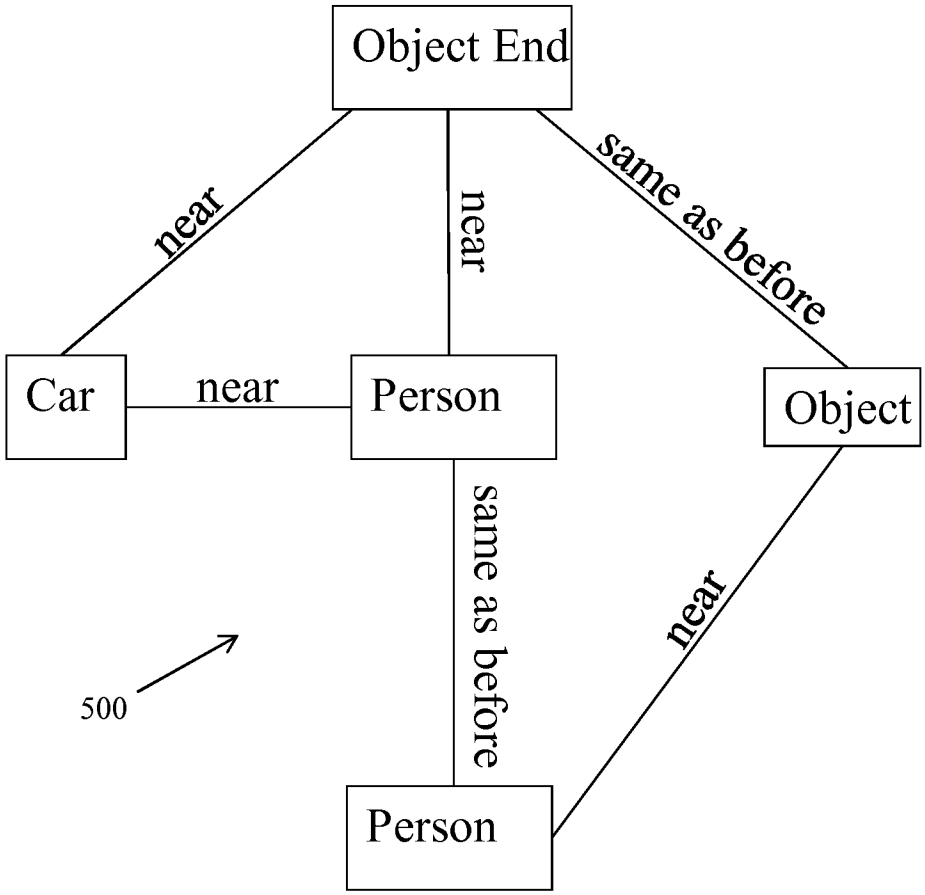

Object End
near
near
same as before
Car
near
Person
Object
same as before
near
500
Person

FIG. 5

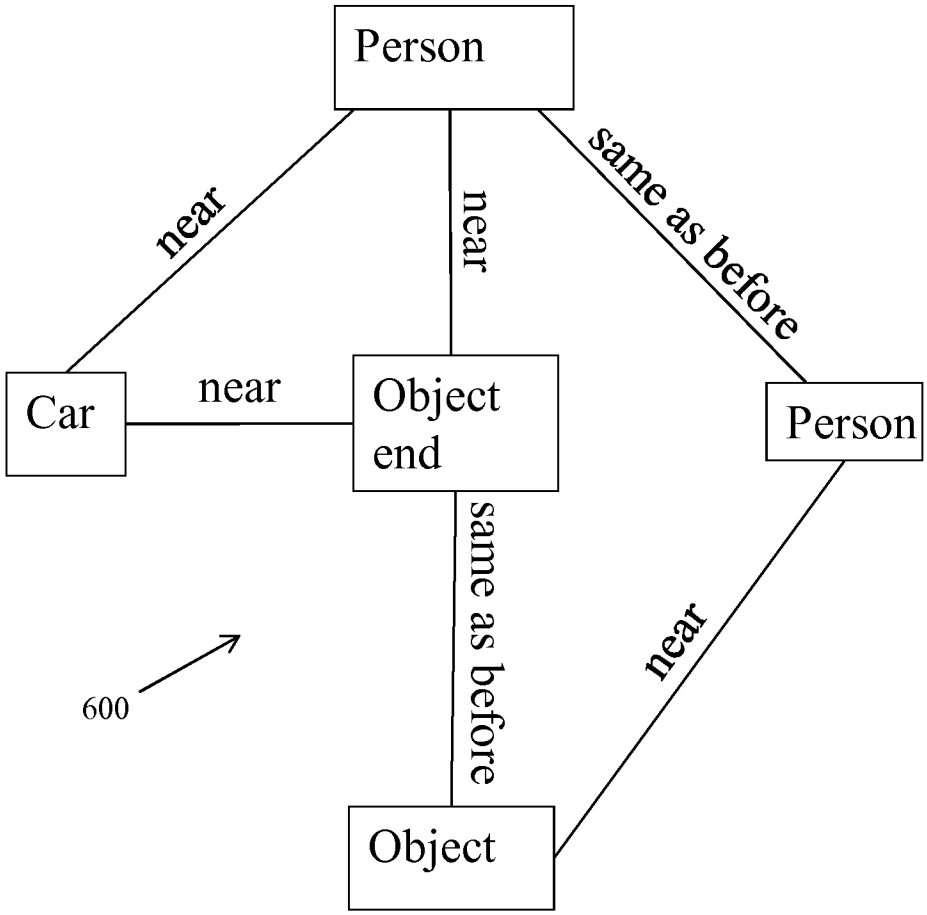

Person
near
near
same as before
Car
near
Object
end
Person
same as before
near
600
Object

FIG. 6

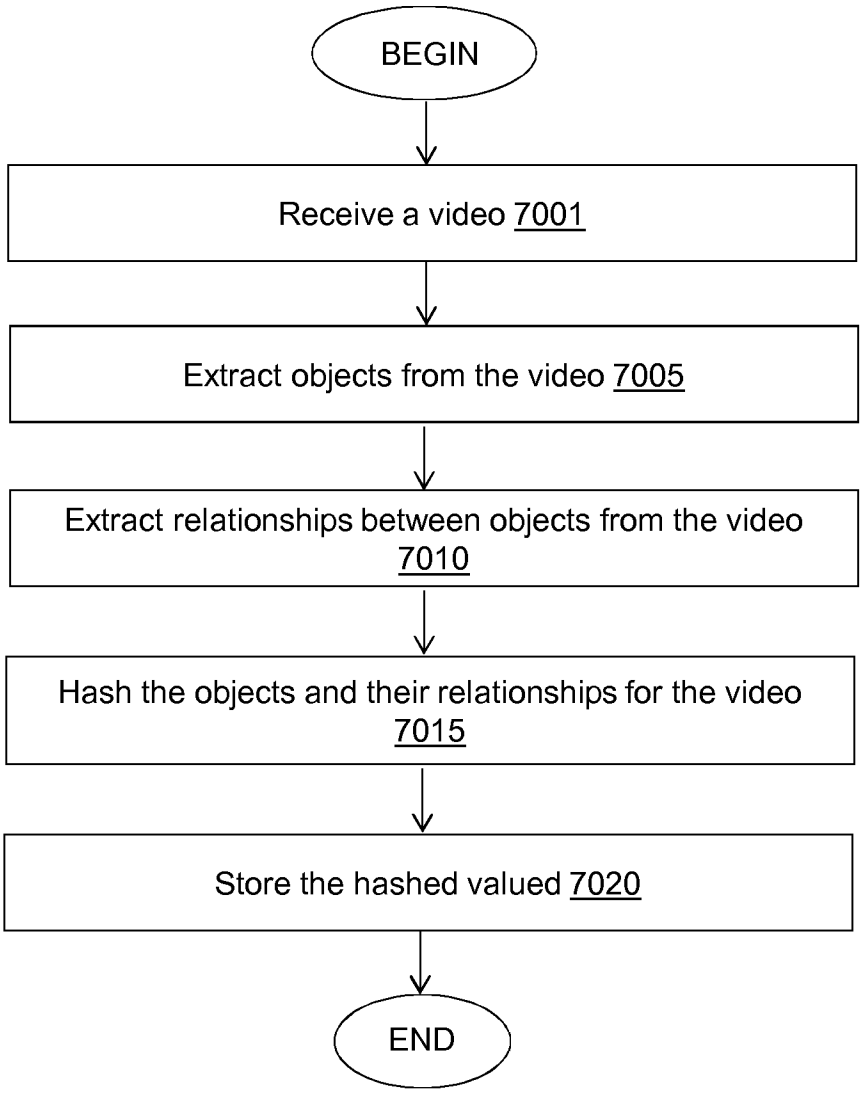

BEGIN
Receive a video 7001
Extract objects from the video 7005
Extract relationships between objects from the video 7010
Hash the objects and their relationships for the video 7015
Store the hashed valued 7020
END

FIG. 7

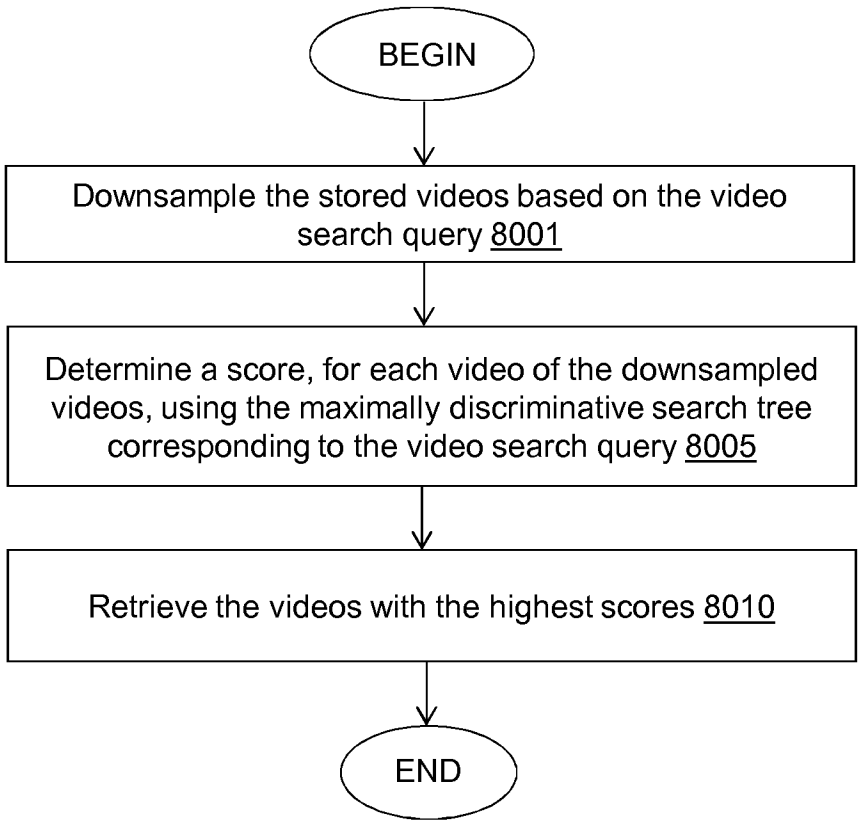

BEGIN
Downsample the stored videos based on the video search query 8001
Determine a score, for each video of the downsampled videos, using the maximally discriminative search tree corresponding to the video search query 8005
Retrieve the videos with the highest scores 8010
END

FIG. 8

Algorithm 1 Create Matching Graph

```
procedure CREATE MATCHING GRAPH(T, C, K_v(v_1, v_2), K_e(v_1, v_2, v_3, v_4))
    H = G(V^h, E^h) ← 0
    Iterate from root to leaves
    for all v^t ∈ V^t do
        Compute the matches to this vertex
        Nv_t ← (v_t, v_c) where K_v(v_t, v_c) > τ_v
        if Parent(v_t) ≠ ∅ then
            E ← ∅
            for all (v_{t_p}, v_{c_p}) ∈ N_{Parent(v_t)} do
                E ← E ∪ (v_t, v_c) where K_e(v_{t_p}, v_{c_p}, v_t, v_c) > τ_e
            end for
            N_{v_t} ← N_{v_t} ∩ E
            V^h ← E^h ∪ N_{v_t}
            E^h ← E^h ∪ E
        else
            V^h ← V^h ∪ N_{v_t}
        end if
    end for
end procedure
```

FIG. 9

Algorithm 2 Solve for Optimal Matches

procedure OPTIMIZE MATCHES(T, H)
  $Score(v) \triangleq K_v(v[0], v[1])$
  $Score(v_1, v_2) \triangleq K_e(v_1[0], v_1[1], v_2[0], v_2[1])$
  Iterate from leaves to root
  for all $v_t \in T$ **do**
    if $Parent(v_t) \neq \emptyset$ **then**
      for all $v \in V^h$ where $v[0] == Parent(v_t)$ **do**
        $Score(v) += max_{c \in Children(v)}(Score(c) + Score(v, c))1(c[0] == v_t)$
      end for
    end if
  end for
end procedure

FIG. 10

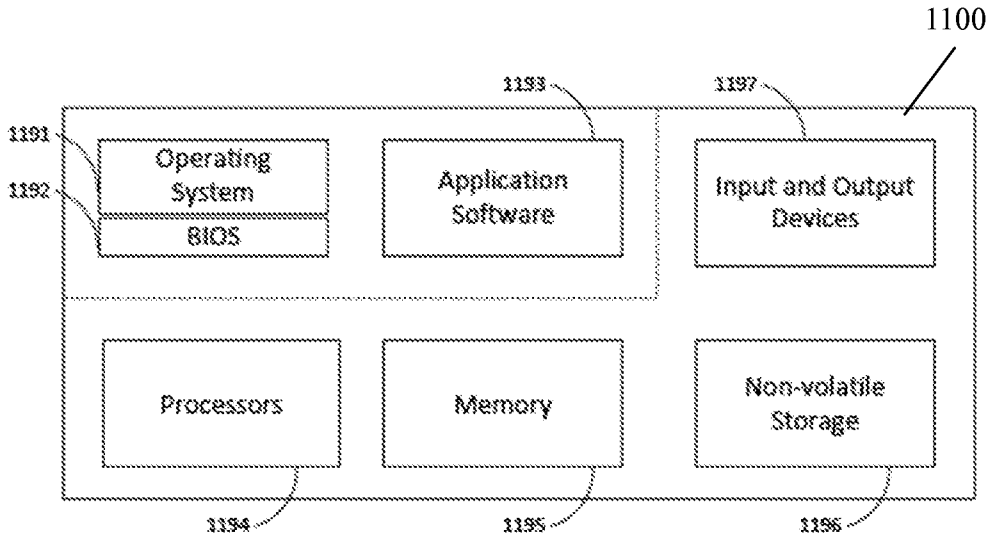

1100
1193
1197
1191
1192
Operating System
BIOS
Application Software
Input and Output Devices
Processors
Memory
Non-volatile Storage
1194
1195
1196

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/061949

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 7/06 (2015.01)
CPC - G06F 7/06 (2015.12)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - G06F 7/06; G06F 17/30; G06K 9/46; H04N 5/14 (2015.01)
CPC - G06F 7/06; G06F 17/30; G06K 9/46; H04N 5/14 (2015.12)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
382/224; 707/705, 706 (Keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Orbit, Google Patents, Google Scholar, Google
Search terms used: video search, engine, video segment, object, relationship, display, query

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 8,254,699 B1 (ZHAO et al) 28 August 2012 (28.08.2012) entire document	1-3, 8, 14, 15, 20 --- 4-7, 9-13, 16-19, 21, 22
Y	US 20140201769 A1 (NEUMEIER et al) 17 July 2014 (17.07.2014) entire document	4, 5
Y	US 20130151562 A1 (FUJII et al) 13 June 2013 (13.06.2013) entire document	6
Y	US 20090292685 A1 (LIU et al) 26 November 2009 (26.11.2009) entire document	7, 12, 13, 21
Y	US 20070239694 A1 (SINGH et al) 11 October 2007 (11.10.2007) entire document	9, 12, 13
Y	US 20110170596 A1 (SHI et al) 14 July 2011 (14.07.2011) entire document	10, 11, 22
Y	US 20140136536 A1 (MICROSOFT CORPORATION) 15 May 2014 (15.05.2014) entire document	16
Y	US 20130016213 A1 (SOLUTIONS XYZ LLC et al) 17 January 2013 (17.01.2013) entire document	17, 18
Y	US 20090175538 A1 (BRONSTEIN et al) 09 July 2009 (09.07.2009) entire document	19

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
05 January 2016	27 JAN 2016
Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774